

Skills

Languages
C#
Visual C# .Net
Visual C++
C++
Java
Visual Basic .Net
ASP.Net

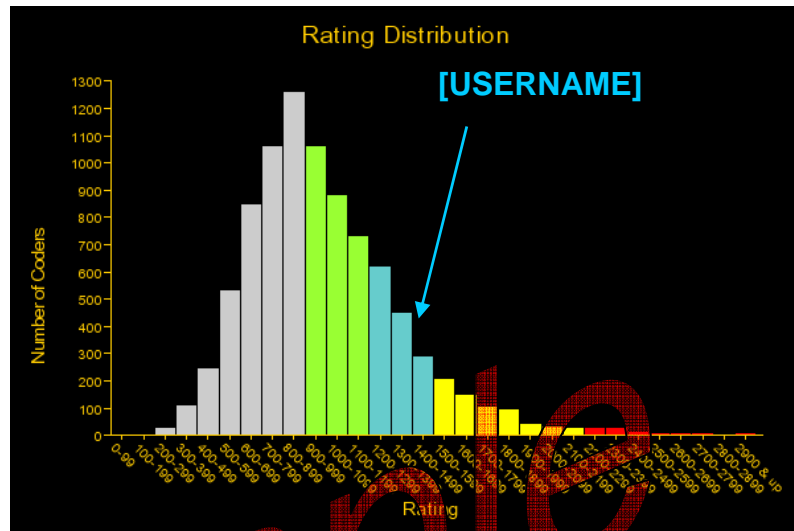
Technologies
OO Concepts
MFC
IIS
COM/DCOM
ADO.Net

Databases
MS SQL Server
Oracle
MySQL

Operating Systems
Windows 2000
Linux
Unix

Rating Position Among Members

Coder Ranking (Based on X Rated Events)			
Rating	1364	Division	1
Rank	1229 out of 9249	Rank %	87 th Percentile



Lifetime Competition Statistics

Solutions Submitted	
Submission Ratio	16 out 23 (70%)
Submission Success Ratio	11 out 16 (69%)

Challenges against solutions of other competitors	
Total Challenged	2
Challenge Success Ratio	50%

Member Profile for [USERNAME]

Code Sample For:	Single Round Match 215 – Level Two Problem	
	Division Statistics	[USERNAME]'s Statistics
Average Elapsed Time to Submit	24:26	36:11
Total Submission %	83%	Submitted
Successful Submission %	65%	Successful

Problem Statement

An entry in a Thesaurus is a list of words that are all synonyms. Each entry contains no duplicates within it. It is possible that two entries might have some common words, but the editors (who are somewhat cheap) have decided that if any two entries have 2 or more words in common then they should be combined into a single entry.

This editing process may produce new entries which can be combined. The final Thesaurus must contain no pair of entries that have 2 or more words in common. Of course, each entry must contain no duplicates.

Create a class Thesaurus that contains a method `edit` that is given a `String[] entry`, the entries in the original Thesaurus. The method returns the edited Thesaurus as a `String[]`. Each element of `entry` has no leading or trailing spaces and has its words separated by a single space. Each element of the return should also have no leading or trailing spaces and have its words separated by a single space. In addition, the words within each element of the return must be in alphabetical order, and the elements in the return must appear in alphabetical order.

Definition

Class: Thesaurus
Method: edit
Parameters: String[]
Returns: String[]
Method signature: String[] edit(String[] entry)
(be sure your method is public)

Constraints

-`entry` will contain between 1 and 50 elements inclusive.

[TOPCODER]

Member Profile for [USERNAME]

- Each element of **entry** will contain between 1 and 50 characters inclusive.
- Each element of **entry** will consist of 1 or more "words" separated by single spaces.
- Each element of **entry** will contain no leading or trailing spaces.
- Each "word" will consist of 1 or more lowercase letters 'a'-'z'-No element of **entry** will contain two identical words.

Examples

0)

```
{"ape monkey wrench", "wrench twist strain"}
```

```
Returns: { "ape monkey wrench", "strain twist wrench" }
```

These two entries have only one common word so they cannot be combined. After rearranging the words within each entry to put the words into alphabetical order, the first entry is first alphabetically.

1)

```
{"ape monkey wrench", "wrench twist strain", "monkey twist frugue"}
```

```
Returns: { "ape monkey wrench", "frugue monkey twist", "strain twist wrench" }
```

No entries could be combined, but two had to be arranged, and the order was changed.

2)

```
{"ape monkey wrench", "wrench twist strain", "monkey twist frugue strain"}
```

```
Returns: { "ape frugue monkey strain twist wrench" }
```

The first two entries could not be combined, but the last two could. After they were combined, the first entry shared both "wrench" and "monkey" with the new combined entry, so we ended up with just one entry.

3)

```
{"point run score", "point dot", "cut run tear score", "cut valley", "cute pretty"}
```

```
Returns: { "cut point run score tear", "cut valley", "cute pretty", "dot point" }
```

This problem statement is the exclusive and proprietary property of TopCoder, Inc. Any unauthorized use or reproduction of this information without the prior written consent of TopCoder, Inc. is strictly prohibited. © 2004, TopCoder, Inc. All rights reserved.

Member Profile for [USERNAME]

[USERNAME]'s Submission

```
using System;
using System.Text;
using System.Collections;

public class Thesaurus
{
    public void runTest()
    {
        string[] reVal;
        //bool pass = false;

        Thesaurus tstObj = new Thesaurus();

        reVal =
            tstObj.edit(new string[] {"point dot run score", "point dot", "point dot cut
run tear score valley", "cut valley", "cute pretty"});

        Console.Out.WriteLine(@"new string[] { "cut point run score tear", "cut va
lley", "cute pretty", "dot point" }");
        Console.Out.WriteLine("");

        reVal =
            tstObj.edit(new string[] {"ape monkey wrench", "wrench twist strain"});

        Console.Out.WriteLine(@"new string[] { "ape monkey wrench", "strain twist wren
ch" }");
        Console.Out.WriteLine("");

        reVal =
            tstObj.edit(new string[] {"ape monkey wrench", "wrench twist strain", "monkey tw
ist frugue"});

        Console.Out.WriteLine(@"new string[] { "ape monkey wrench", "frugue monkey twi
st", "strain twist wrench" }");
        Console.Out.WriteLine("");

        reVal =
            tstObj.edit(new string[] {"ape monkey wrench", "wrench twist strain", "monkey tw
ist frugue strain"});

        Console.Out.WriteLine(@"new string[] { "ape frugue monkey strain twist wrench" }
");
        Console.Out.WriteLine("");

        reVal =
            tstObj.edit(new string[] {"point run score", "point dot", "cut run tear score", "cu
t valley", "cute pretty"});

        Console.Out.WriteLine(@"new string[] { "cut point run score tear", "cut valley
", "cute pretty", "dot point" }");
```

Member Profile for [USERNAME]

```
Console.Out.WriteLine("");
}

public string[] edit(string[] entry)
{
    string[][] input = new string[entry.Length][];

    for (int i=0; i<entry.Length; i++)
    {
        input[i] = entry[i].Split(' ');
    }

    string[][] re = orderAndCombine(input);

    ArrayList ret = new ArrayList();

    for (int i=0; i<re.Length; i++)
    {
        string total = "";
        for (int j=0; j<re[i].Length; j++)
        {
            total += re[i][j] + " ";
        }
        total = total.Trim();
        ret.Add(total);
    }

    string[] result = (string[]) ret.ToArray(typeof(string));
    Array.Sort(result);

    return result;
}

private string[][] orderAndCombine(string[][] input)
{
    //Sort them first
    for (int i=0; i<input.Length; i++)
    {
        Array.Sort(input[i]);
    }

    for (int i=0; i<input.Length-1; i++)
    {
        for (int j=i+1; j<input.Length; j++)
        {
            // Check if we have more than two match
            if (hasMoreThanTwoMatch(input[i], input[j]))
            {
                string[][] newInput = new string[input.Length-1][];
                int x =0;
            }
        }
    }
}
```

Sample

Member Profile for [USERNAME]

```
        newInput[x++] = combineTwoArray(input[i], input[j]);

        for (int z=0; z<input.Length; z++)
        {
            if (z!=i && z!=j)
            {
                newInput[x++] = input[z];
            }
        }

        return orderAndCombine(newInput);
    }
}

return input;

//      int[,] hitTable = new int[input.Length, input.Length];
//      hitTable.Initialize();
//
//      Hashtable words = new Hashtable();
//
//      for (int i=0; i<input.Length; i++)
//      {
//          for (int j=0; j<input[0].Length; j++)
//          {
//              string thisWord = input[i][j];
//
//              if (!words.ContainsKey(thisWord))
//              {
//                  words.Add(thisWord, new ArrayList());
//              }
//
//              ArrayList lst = (ArrayList) words[thisWord];
//              lst.Add(i);
//
//              if (lst.Count > 1)
//              {
//                  foreach (
//
//              }
//          }
//      }
//
//      foreach (ArrayList itm in words)
//      {
//
//      }
}

private string[] combineTwoArray(string[] strA, string[] strB)
{
    Hashtable ht = new Hashtable();

    for (int i=0; i<strA.Length; i++)
    {
        if (!ht.ContainsKey(strA[i])) ht.Add(strA[i], strA[i]);
    }
}
```

sample

Member Profile for [USERNAME]

```
    }  
    for (int i=0; i<strB.Length; i++)  
    {  
        if (!ht.ContainsKey(strB[i])) ht.Add(strB[i], strB[i]);  
    }  
  
    string[] re = new string[ht.Count];  
  
    ht.Values.CopyTo(re, 0);  
  
    return re;  
}  
  
private bool hasMoreThanTwoMatch(string[] strA, string[] strB)  
{  
    int totalMatch = 0;  
  
    for (int i=0; i<strA.Length; i++)  
    {  
        for (int j=0; j<strB.Length; j++)  
        {  
            if (string.Compare(strA[i], strB[j], true) == 0)  
            {  
                totalMatch++;  
                if (totalMatch >= 2)  
                {  
                    return true;  
                }  
            }  
        }  
    }  
  
    return false;  
}  
}
```

Sample

Member Profile for [USERNAME]

[USERNAME]'s Resume

Objective

Work in a dynamic and fast pace development team. Design, develop, and deploy solutions that fit customers' needs.

Experience

Cerqa Supply Chain Management, Austin, Texas 2003-present

Senior Software Developer

- Architect, design, and develop B2B applications utilizing Microsoft .Net Framework, SQL Server, CRM Server, and Biztalk.
- Create and review the company's development methodologies and coding standard.
- Created company's Object Library to encourage object reuse.
- Created company's Code Library to encourage code reuse.
- Object Library and Code Library increases the company's overall development speed and product quality.
- Design and develop company's ERP integration applications.
- Design and develop company's Shipping applications.
- Design and develop company's EDI communication applications.
- Design and develop company's Call Center integration applications.
- Design and develop company's Customer Report Factory application.
- Design and develop company's Customer Portal application.
- Work as a project manager and technology consultant for company's clients.
- Remotely manage and direct clients' .Net and J2EE application development.
- Manage and motivate company's software development team and IT team as acting CTO.
- Conduct performance reviews for developers and IT stuffs.

VI Technology, Austin, Texas 2000-2003

Project Manager / Computer Scientist

- Lead the development of an Enterprise test data management system that is the company's main product.
- Develop solutions with C#, VB.Net, VB, MS SQL Sever, MSDE, COM+, ASP, XML, DHTML, JavaScript, MSMQ, Windows 2000, and Active Director technologies.
- Lead the development of company's purchasing/receiving system.
- Create company's Product application for product number assignment, product BOM(build of material) management, and total cost estimate.
- Developed automated testing systems for communication chips.
- Managed all projects to customer satisfaction. Received "Outstanding Engineering Leadership" Award. Received number of recognitions for outstanding performances.

Computer Scientist 1998-2000

- Software development using C/C++, VB, Perl, SQL, PHP, and Java. Worked in Win32, Linux, and Web environment.

[TOPCODER]

Member Profile for [USERNAME]

- Provide IT support and management for users. Involved in the design of a time-track software package.
- Designed and implemented a Linux server cluster. Received award for outstanding efforts in one of the R&D projects.

EDS, Austin, Texas 1996-1998

Programmer

- Working with clients to develop their business requirements and translating those requirements into solutions.
- Interface with clients, peers, leaders, vendors, and internal support groups. Meet client needs and expectations by designing, constructing, testing, and implementing integrated network, hardware, and software solutions.

University of Texas at Austin, Austin, Texas 1994-1996

Computer Lab Assistant

- Trouble shoots computers and network systems. Answer lab users' questions.
- Converted the lab network system from Netware to Windows NT to reduce operation cost.
- Obtained skill in trouble shooting network systems (10BaseT and 10Base2) and computer stations (Win NT 3.51, Win 3.11, and Mac OS)

Education

M.C.S.E.

November 1999

University of Texas, Austin, Texas

Bachelor of Science, December 1995

- Dean's Honor
- Golden Key Honor Society Faculty

Skills

- Excellent Management skill.
- Expert C#, VB.NET, Java, ASP.NET, VB, ASP, XML, T-SQL and C / C++ programmer.
- Experienced in Microsoft Biztalk, CRM, and Project server.
- Experienced in Win32, Linux, and Web environment.